

The Connected p -Center Problem with Extension

William Chung-Kung Yen and Chien-Tsai Chen

Department of Information Management, Shih Hsin University, Taipei, Taiwan

Email: {ckyen001@ms7.hinet.net, hbk_156@hotmail.com}

Abstract¹

Let $G(V, E, W)$ be a graph n vertices and m edges, where each edge e is associated with a positive distance $W(e)$. The traditional p -Center problem is to locate some kind of facilities at p vertices of G to minimize the maximum distance between any vertex and its nearest facility. This paper proposes a practical constraint: the subgraph induced by the p facility vertices must be connected and the problem is called the Connected p -Center problem. We show that the problem on bipartite graphs is NP-Hard, but $O(n)$ -time solvable on trees. After then, the algorithmic result on trees is extended to the situation that some vertices in V cannot be selected as facility vertices.

Keywords: connected p -center, induced subgraph, NP-Hard, bipartite graph, tree, forbidden vertex

1. Introduction

Client/server architecture has become a basis for networks and distributed systems. Consider the situation that there are resources, e.g., servers, programs, routers, data objects, etc., to be built at some vertices to provide services requested by the clients all over a computer network. If there is no facility at a node u , the clients at u need to move to a nearest node at which there is a facility located to get the required service. These types of applications correspond to the fundamental discrete location problem, the p -Center problem.

Let $G(V, E, W)$ be a graph with n vertices and m edges, where each edge e is associated with a positive distance $W(e)$. For any $Q \subseteq V$, the distance between Q and a vertex $v \notin Q$, is defined as $d(v, Q) = \min_{u \in Q} \{d(v, u)\}$, where $d(x, y)$ is the length of the shortest path between any pair of vertices x and y . Meanwhile, we define $\delta(Q) = \max_{v \in V-Q} \{d(v, Q)\}$. The p -Center problem can be described as follows [5].

The p -Center problem: Given a graph $G(V, E, W)$ and an integer $p \geq 1$, identify a subset $H = \{q_1, \dots, q_p\}$ of V , called a p -center of G , such that $\delta(H)$ is minimized.

Fig. 1 shows an input instance of the 2-Center problem. It is easy to verify that $H = \{v_3, v_6\}$ is a 2-center of this graph such that $\delta(H) = 8$ is minimized.

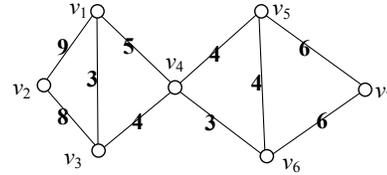


Fig. 1: A graph with distances on edges.

The p -Center problem has wide-area applications to real-world problems. Extensive research effort has been done on it [1, 4, 5, 10]. The problem has been known to be NP-Hard [7]. In [11], the author provided an $O(n)$ -time algorithm for the problem on interval graphs ($p = 1$). The author in [12] extended the result of [11] general $p \geq 1$ and the time-complexity is still $O(n)$. Lan et al. proposed a linear-time algorithm for finding centers on weighted cactus graphs [14]. Frederickson solved this problem on trees in linear-time (without necessarily restricting the location of the facilities to the vertices of the tree) using parametric search [6]. Bspamyatnikh et al. gave an $O(pn)$ -time algorithm for the p -center problem on circular-arc graphs [3]. Hsu et al. presented a general p -facility location problem on the real line with unimodal distance functions and designed an $O(pn^2)$ -time algorithm for it [8]. Kariv and Hakimi designed an $O(n^{2p+1} \log n / (p-1)!)$ -time algorithm for the p -center problem on general graphs [10]. Tamir showed that the p -center problems on weighted and unweighted networks can be solved in $O(n^p m^p \log^2 n)$ -time and $O(n^{p-1} m^p \log^3 n)$ -time, respectively [13]. In addition, some works have been done on approximation algorithms for finding p -centers on graphs [2, 9].

This paper proposes a very practical constraint: the subgraph induced by the p facility vertices must be connected. The issue is very practical and essential when some facility vertex, say v , has very heavy task load. No routing overhead will occur for other facility vertices to support the services provided by v if the p facility vertices induce a connected subgraph. A p -center of this type is called a *connected p -center* hereafter. In our concerns, it is reasonable and natural to assume that $p \geq 2$ and the number of the vertices in

¹ This research was supported by National Science Council, Taiwan, under the contract number NSC94-2213-E-128-005.

G is at least 2, i.e., $n \geq 2$, in the rest of this paper.

The Connected p -Center Problem (The CpC problem): Given a graph $G(V, E, W)$ and an integer $p \geq 2$, identify a connected p -center $Q = \{q_1, \dots, q_p\}$ of V such that $\delta(Q)$ is minimized. Q is called an *optimal solution* of the CpC problem on G hereafter.

Let us examine Fig. 1 again. $Q = \{v_3, v_4\}$ is a connected 2-center such that $\delta(Q) = 9$ is minimized.

2. The CpC Problem on Bipartite Graphs

A graph $G(V, E)$ is a *bipartite graph* if V can be partitioned into two disjoint sets I and J such that if $(u, v) \in E$, then either $(u \in I \text{ and } v \in J)$ or $(u \in J \text{ and } v \in I)$. A bipartite graph will be denoted by $G(I \cup J, E)$ hereafter. The following decision problem and NP-complete problem [7] are introduced.

The CpC decision problem: For a graph $G(V, E, W)$, an integer $p \geq 2$, and a positive constant λ , determine whether there exists a connected p -center Q of G such that $\delta(Q) \leq \lambda$ or not?

The Monotone Satisfiability problem (The MSAT problem): Given a set of Boolean clauses in the conjunctive normal form in which each clause contains either only positive literals, say x_i 's, or only negative literals, say \bar{x}_i 's, determine whether the given Boolean formula is satisfiable or not?

Lemma 1. The CpC decision problem on bipartite graphs is NP-complete.

Proof: It is clear that the CpC decision problem is a NP problem. Suppose that there is an instance of the MSAT problem with p variables x_1, \dots, x_p , $p \geq 2$, and r clauses c_1, \dots, c_r . Assume that $C^{(P)} = \{c_1, \dots, c_\alpha\}$ and $C^{(N)} = \{c_{\alpha+1}, \dots, c_r\}$ represent the set of clauses containing positive literals and negative literals, respectively. If one of $C^{(P)}$ and $C^{(N)}$ is empty, then the MSAT problem is trivial. Hence, we assume that both $C^{(P)}$ and $C^{(N)}$ are nonempty.

Denote $A = \{a_1, \dots, a_p\}$, $B = \{b_1, \dots, b_p\}$, $X = \{x_1, \dots, x_p\}$, and $\bar{X} = \{\bar{x}_1, \dots, \bar{x}_p\}$. A graph $G(V, E)$ can be constructed by the following transformation rules for any given positive constant λ : (1) $V = C^{(P)} \cup C^{(N)} \cup X \cup \bar{X} \cup A \cup B$. (2) $E = \{(c_j, \underline{x}_i) \mid \text{for all } c_j \text{ contains } \underline{x}_i, 1 \leq j \leq \alpha\} \cup \{(c_j, x_i) \mid \text{for all } c_j \text{ contains } x_i, \alpha + 1 \leq j \leq p\} \cup \{(x_i, a_i), (a_i, b_i), (b_i, x_i) \mid 1 \leq i \leq p\} \cup \{(x_s, x_t) \mid \text{for all } 1 \leq s \neq t \leq p\}$. (3) $W(x_i, a_i) = W(a_i, b_i) = W(b_i, x_i) = (1/3)\lambda$, $1 \leq i \leq p$. (4) $W(e) = \lambda$, for all other edges e .

It is very clear that the time-complexity to construct the graph G is polynomial with respect to p and r , and G is a bipartite graph in which $I = C^{(P)} \cup \bar{X} \cup A$ and $J = C^{(N)} \cup X \cup B$.

Assume that there is a truth assignment for the

Boolean formula. We can assume that $x_1 = \dots = x_k = \text{TRUE}$ and $\bar{x}_{k+1} = \dots = \bar{x}_p = \text{FALSE}$. Let $Q = \{x_1, \dots, x_k, x_{k+1}, \dots, x_p\}$. Since both $C^{(P)}$ and $C^{(N)}$ are nonempty, we must have $|Q \cap X| \geq 1$ and $|Q \cap \bar{X}| \geq 1$. Meanwhile, each x_s is adjacent to \bar{x}_t , for all $s \neq t$. The subgraph induced by Q must be connected. We can easily verify that $\delta(Q) \leq \lambda$, i.e., Q a connected p -center of G such that $\delta(Q) \leq \lambda$.

Suppose that Q is connected p -center of G such that $\delta(Q) \leq \lambda$. We claim that for each pair x_j and \bar{x}_j , at least one of them belongs to Q . Let there exist j such that $x_j \notin Q$ and $\bar{x}_j \notin Q$. Since Q induces a connected subgraph and $p \geq 2$, we have $a_j \notin Q$ and $b_j \notin Q$. Let u_j and v_j be the nearest vertices in Q corresponding to a_j and b_j , respectively. The construction rules of G imply that $d(u_j, a_j) > \lambda$ and $d(v_j, b_j) > \lambda$. It contradicts to the fact that $\delta(Q) \leq \lambda$.

Since $|Q| = p$ and Q contains at least one vertex of each pair $\{x_j, \bar{x}_j\}$, $1 \leq j \leq p$, this can imply that Q contains exactly one vertex in each pair $\{x_j, \bar{x}_j\}$, $1 \leq j \leq p$. In addition, both $C^{(P)}$ and $C^{(N)}$ are nonempty, we must have $|Q \cap X| \geq 1$ and $|Q \cap \bar{X}| \geq 1$. This means that the nearest vertex in Q of each clause c_j is either x_t or \bar{x}_t , for some t . Setting the literals corresponding to the vertices in Q to be TRUE certainly satisfies the Boolean formula. ■

Theorem 1. The CpC problem on bipartite graphs is NP-Hard, $p \geq 2$.

3. An $O(n)$ -time Algorithm on Trees

We prove the following lemma to identify the starting vertex for obtaining an optimal solution of a tree T .

Lemma 2. Given a tree T , suppose that u and v is a pair of vertices such that $d(u, v) \geq d(x, y)$, for all $x, y \in V$. Let m be the middle point of the shortest path from u to v , i.e., $d(u, m) = d(m, v) = (1/2) * d(u, v)$ and r be a vertex nearest to m . If Q is a connected p -center of T such that $r \in Q$, then, $\delta(Q) \leq \delta(H)$, for all connected p -center H in which $r \notin H$.

Proof: By the definition of vertex r , the middle point m must lie on some edge incident with r . Here we just assume that m lies within the edge (r, x_j) as shown in Fig. 2, where r is assigned as the root of T and x_1, \dots, x_k are the children of r . The other case can be easily handled using the same way.

Denote $V(T(x_j))$ as the vertex-set of the subtree rooted at x_j , $1 \leq j \leq k$. It is easy to see that $\{r, x_j\} \subseteq Q$. This implies that $\delta(Q) \leq \max\{d(v, x_j), d(u, r)\} < d(v, m) = (1/2)d(u, v)$. Since $r \notin H$, by the definitions of trees and connected p -centers, we must have $H \subseteq V(T(x_j))$, for some subtree $T(x_j)$. The following two cases will be handled, respectively.

Case 1. $H \subseteq V(T(x_j))$: We can derive that $\delta(H) \geq d(u, x_j) = d(u, x_{j+1}) + d(x_{j+1}, m) + d(m, x_j) = (1/2) * d(u, v)$.

$d(u, v) + d(m, x_j)$. This implies that $\delta(H) \geq \delta(Q)$.

Case 2. $H \subseteq V(T(x_t))$, $1 \leq t \leq k$, $t \neq j$: H lies on the subtree $T(x_t)$ in this case. We can easily prove that $\delta(H) \geq d(v, r) + d(r, x_t) = (1/2)d(u, v) + d(m, x_t)$. This derives that $\delta(H) \geq \delta(Q)$. ■

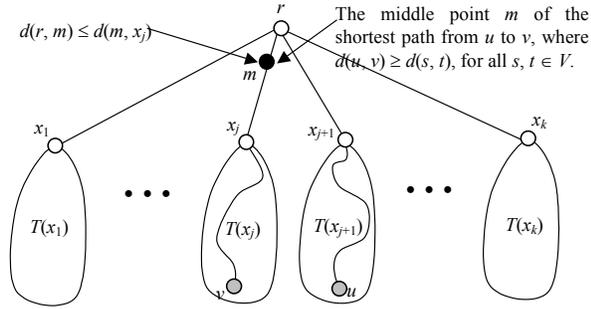


Fig. 2: The middle point m of the shortest path from u to v lies within (r, x_j) .

The task left is to identify which vertices in $V - \{r\}$ should be included to obtain an optimal solution Q . Denote the tree T as $T(r)$.

Definition 1. For each $x \in V - \{r\}$, $\text{par}(x)$ represents the parent vertex of x and $\text{ace}(x)$ represents the set of all ancestors of x . Meanwhile, if x is not a leaf, then $\text{Children}(x)$ denotes the set of its child vertices.

Definition 2. For each vertex z of $T(r)$, $\mu(z)$ is defined as follows: (1) $\mu(z) = W(z, \text{par}(z))$, if z is a leaf. (2) $\mu(z) = \max_{t \in \text{Children}(z)} \{\mu(t)\} + W(z, \text{par}(z))$, if z is not the root r and z is non-leaf. (3) $\mu(r) = \max_{t \in \text{Children}(r)} \{\mu(t)\}$.

Lemma 3. $\mu(r) \geq \mu(z)$, for all $z \in V$, and given any vertex $x \in V - \{r\}$, $\mu(y) > \mu(x)$, for all $y \in \text{ace}(x)$.

Lemma 4. Let $H = \{h_1, \dots, h_\alpha\}$ such that $\mu(h_1), \dots, \mu(h_\alpha)$ are the first α largest numbers among $\{\mu(z) \mid z \in V\}$, $\alpha \geq 2$. Then, H forms a subtree of $T(r)$, i.e., H is a connected α -center of $T(r)$ and $\delta(H) = \mu(h_{\alpha+1})$, where $\mu(h_{\alpha+1})$ denotes the $(\alpha + 1)$ st number among $\{\mu(z) \mid z \in V\}$.

Proof. Suppose that H is not a connected α -center of $T(r)$. Let j , $2 \leq j \leq \alpha$, be the smallest number such that $H^* = \{h_1, \dots, h_j\}$ forms a subtree but h_{j+1} is not adjacent to any vertex in H^* . Since we only consider connected graphs, Lemma 3 implies that $\text{par}(h_{j+1}) \notin H$. But Lemma 3 states that $\mu(\text{par}(h_{j+1})) > \mu(h_{j+1})$ and this also implies that the first α largest numbers among $\{\mu(z) \mid z \in V\}$ must include $\mu(\text{par}(h_{j+1}))$, i.e., $\text{par}(h_{j+1}) \in H$. A contradiction occurs.

Finally, according to the definition of $\mu(z)$ of each vertex z , it is easy to verify that $\delta(H) = \max_{y \in V-H} \{\mu(y)\} = \mu(h_{j+1})$. ■

Corollary 1. Let $Q = \{q_1, \dots, q_p\}$ such that $\mu(q_1), \dots, \mu(q_p)$ are the first p largest numbers among $\{\mu(z) \mid z \in V(T(r))\}$. Then, Q is an optimal solution of $T(r)$.

The discussion so far can verify the correctness of the following algorithm.

Algorithm Connected p -Center on Trees

Input: A tree T such that each edge e is associated a positive distance $W(e)$ and a positive integer $p > 1$.

Output: A connected p -center Q such that $\delta(Q)$ is minimized.

Step 1: Find a pair of vertices u and v such that $d(u, v) \geq d(x, y)$, for all pairs of vertices x and y of T ;

Step 2: Identify the root r as stated above;

Step 3: Compute $\mu(z)$ for each vertex z ;

Step 4: $Q = \{q_1, \dots, q_p\}$ such that $\mu(q_1), \dots, \mu(q_p)$ are the first p largest numbers among $\{\mu(z) \mid z \in V\}$;

Step 5: Return Q ;

End Connected p -Center on Trees

Theorem 2. The CpC problem on trees can be solved in $O(n)$ -time.

Proof. In the above algorithm, Step 1 through Step 3 can be done in $O(n)$ -time [5]. Step 4 involves finding the p th largest number among $\{\mu(z) \mid z \in V\}$ and can be done in $O(n)$ -time. Thus, the total time-complexity is $O(n)$. ■

4. Extension to Trees with Forbidden Vertices

In real-world systems, some vertices may not be suitable as center vertices due to function failure or some practical constraints such as capacity and processing ability. We use F to represent such vertices and called them *forbidden vertices*. The section will extend the results of Section 3 to the situation that the vertices in $F \subseteq V$ cannot be included in any connected p -center H of a tree T .

Definition 3. For each vertex z of T , $\Delta(z)$ is defined as follows: (1) $\Delta(z) = 0$, if $z \in F$. (2) $\Delta(z) = 1$, if z is a leaf and $z \notin F$. (3) $\Delta(z) = \sum_{y \in \text{Children}(z)} \Delta(y) + 1$, if z is not a leaf and $z \notin F$.

Fig. 3 shows a tree with forbidden vertices (black vertices) and the numbers within braces are the Δ values of each vertex. Any vertex can be assigned as the root. For example, $\Delta(y_4) = \sum_{u \in \text{Children}(y_4)} \Delta(u) + 1 = \Delta(y_6) + \Delta(y_7) + 1 = 3$. The vertex a is the root of T and $\Delta(a) = \Delta(y_1) + \Delta(y_2) = 3$.

Lemma 5. Given an integer $p \geq 2$, let T_1, \dots, T_k be the disjoint subtrees obtained via deleting all vertices in F . The following properties hold. (1) $|V(T_j)| < p$ iff $\Delta(u) < p$, for all $u \in T_j$. (2) Let H be any connected p -center of T such that $H \cap F = \emptyset$ and T^* be the union of the subtrees satisfying the property (1). Then, $H \cap V(T^*) = \emptyset$.

Lemma 5 implies that all vertices in T^* can be viewed as additional forbidden vertices. For example, in Fig. 3, when $p = 3$, $\{y_3, y_5\}$, $\{y_{10}\}$, and $\{y_{11}\}$ can be viewed as additional forbidden vertices. Therefore, the set of all forbidden vertices is now

shown in Fig. 4 (black vertices). In the rest of this section, we assume that all vertices in T^* are included into forbidden vertices. The following lemma can be proved using similar technique in Lemma 2.

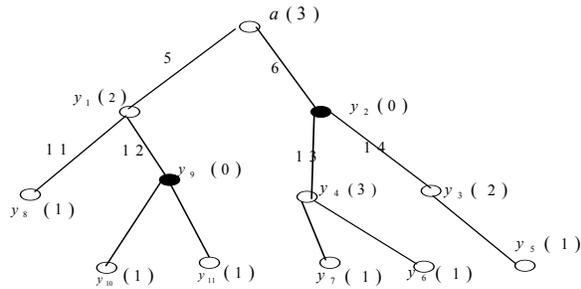


Fig. 3: A tree with forbidden vertices $F = \{y_2, y_9\}$.

Lemma 6. Given a tree T with forbidden set F , let u and v is a pair of vertices such that $d(u, v) \geq d(x, y)$, for all pairs $x, y \in V$. Let m be the middle point of the shortest path from u to v and r be a non-forbidden vertex nearest to m (must exist such vertex). Suppose that Q is a connected p -center of T such that $r \in Q$ and $Q \cap F = \emptyset$. Then, $\delta(Q) \leq \delta(H)$, for all connected p -center H in which $r \notin H$ and $H \cap F = \emptyset$.

After finding the starting vertex, it is easy to verify that the remaining task can be achieved using similar techniques in Section 3.

Theorem 3. The CpC problem on trees with forbidden vertex-set F can be solved in $O(n)$ -time.

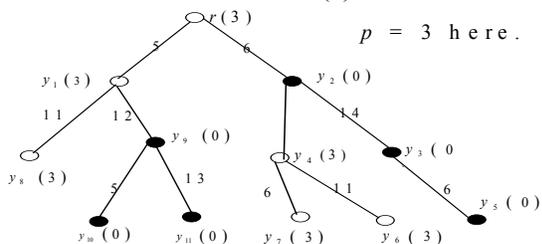


Fig. 4: Adding additional forbidden vertices to Fig. 3.

5. Conclusions

This paper addressed the Connected p -Center problem on graphs. It can be viewed as a practical variant of the traditional p -Center problem. We showed that the problem is NP-Hard on bipartite graphs and $O(n)$ -time solvable on trees. Finally, the algorithmic result was extended to the situation that the input tree has forbidden vertices and time-complexity is still $O(n)$.

In the future, it is practical and meaningful to extend our algorithms to other classes of graphs such as cactus graphs and planar graphs. Meanwhile, identifying other variants of the p -Center problem is a very important issue. For example, restricting that the p -centers must be “total”, i.e., the subgraph induced by the p -centers has no isolated vertices.

References

- [1] F. Abdelaziz, “1-center problem on the plane with uniformly distributed demand points,” *Operations Research Letters*, Vol. 34, Iss. 3, 264-268, 2006.
- [2] J. Bar-Ilan and D. Peleg, “Approximation algorithms for selecting network centers,” *Proc. of Workshop on Algorithms and Data Structures*, 343-354, 1991.
- [3] S. Bespamyatnikh, B. Bhattacharya, M. Keil, D. Kirkpatrick, and M. Segal, “Efficient algorithms for centers and medians in interval and circular-arc graphs,” *Networks*, Vol. 39, 144–152, 2002.
- [4] R. E. Burkard and H. Dollani, “Center problems with pos/neg weights on trees,” *European Journal of Operational Research*, Vol. 145, Iss. 3, 483-495, 2003.
- [5] M. S. Daskin, *Networks and Discrete Location, Models, Algorithms, and Applications*, John Wiley & Sons, Inc., New York, 1995.
- [6] G. Frederickson, “Parametric search and locating supply centers in trees,” *Proc. of Workshop on Algorithms and Data Structures*, 299–319, 1991.
- [7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Bell Laboratories, Murray Hill, Freeman & Co., N. J., 1979,
- [8] V. N. Hsu, T. J. Lowe, and A. Tamir, “Structured p -facility location problems on the line solvable in polynomial time,” *Operations Research Letters*, Vol. 21, 159-164, 1997.
- [9] D. Hochbaum and D. B. Shmoys, “A unified approach to approximation algorithms for bottleneck problems,” *Journal of the ACM*, Vol. 33, 533-550, 1986.
- [10] O. Kariv and S. L. Hakimi, “An algorithmic approach to network location problems I: the p -centers,” *SIAM Journal of Applied Mathematics*, Vol. 37, 514–538, 1979.
- [11] S. Olariu, “A simple linear-time algorithm for computing the center of an interval graph,” *International Journal of Computer Mathematics*, Vol. 24, 121–128, 1990.
- [12] T. C. E. Chenga, L. Kang, and C. T. Ng, “An improved algorithm for the p -center problem on interval graphs with unit lengths,” *Computers & Operation Research*, to appear.
- [13] A. Tamir, “Improved complexity bounds for center location problems on networks by using dynamic data structures,” *SIAM Journal of Discrete Mathematics*, Vol. 1, 377–396, 1988.
- [14] Y-F Lan, Y-L Wang, and H. Suzuki, “A linear-time algorithm for solving the center problem on weighted cactus graphs,” *Information Processing Letters*, Vol. 71, 205-212, 1999.